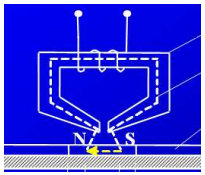
计算机组成原理

计算机组成原理课程组
（刘旭东、高小鹏、肖利民、牛建伟、栾钟治）

第八讲：虚拟存储系统

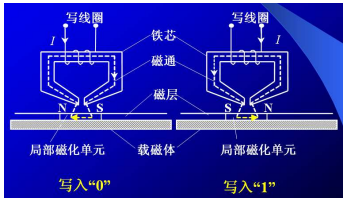
1.1 磁记录编码方式 —— 磁表面存储原理

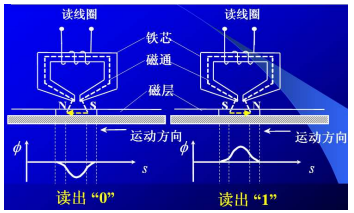
- 磁表面存储器：磁介质 + 磁头
- 磁介质
  - 片基（载体）：塑料（软盘），金属（硬盘）
  - 磁记录材料：极细的γ—Fe_2O_3颗粒，涂（或喷射）在盘面上，形成细密、均匀、光滑的磁膜，硬磁材料
  - 通过磁性材料的磁化状态来记录0和1信息
- 磁头
  - 磁头：是实现电磁转换过程的关键装置，软磁材料，留有间隙的磁性环状物体，上面绕有线圈
  - 当向线圈提供一定方向和大小的电流时，磁头体被磁化，磁头的间隙处产生漏磁，成为信息源
  - 磁头体积小，重量轻，软盘采用接触方式，硬盘采用浮动方式（浮动磁头，薄膜磁头）



1.1 磁记录编码方式 —— 磁表面存储原理

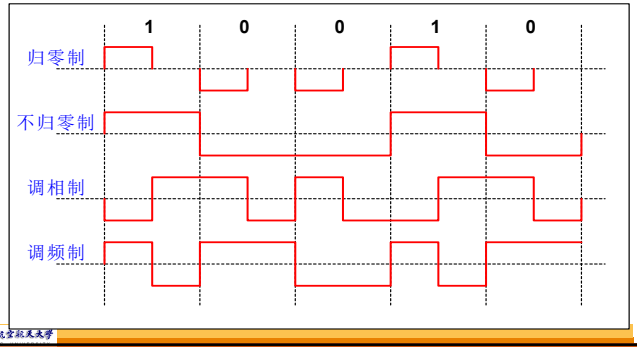
- 磁记录原理：
  - 通过磁头与介质的相对运动完成读写操作
- 写入操作：磁化过程
  - 根据写入代码，确定写入驱动电流的方向，使磁表面被磁化的极性方向不同，以记录“0”和“1”
- 读出操作：感应过程
  - 磁头相对磁化单元做切割磁力线运动，磁化单元的极性决定了感应电势的方向，以此区别“0”和“1”





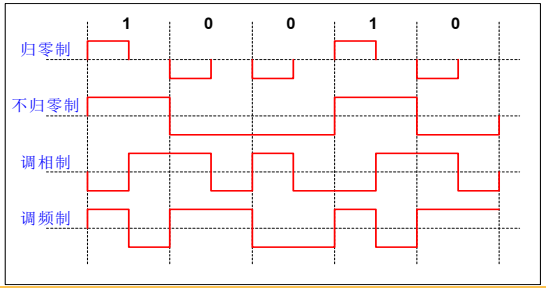
1.1 磁记录编码方式

- **磁记录编码方式：实际上是写入电流的变化方式**
  - 编码方法，即按什么规律，把一连串的二进制信息变换成磁层中的一个序列的磁化翻转状态，且能可靠地用读写电路实现这一转换过程
  - 归零制RZ(正负脉冲电流)、不归零制NRZ(正反向电流)、调相制PM(磁化翻转方向)、调频制FM(磁化翻转频率)



1.1 磁记录编码方式

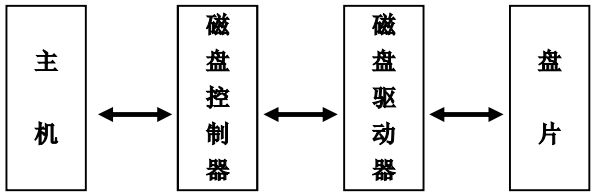
- **评价磁记录编码方式的主要指标**
  - 编码效率：记录一位信息的最大磁化翻转次数的倒数；FM与PM为2（50%），NRZ为1（100%）
  - 自同步能力：能否直接从读出的信号中提取同步信号；NRZ没有自同步能力，PM，FM等都具备自同步能力
  - 可靠性：归零制低，调相制高



第八讲：虚拟存储系统

1.2 硬磁盘存储器 —— 基本结构

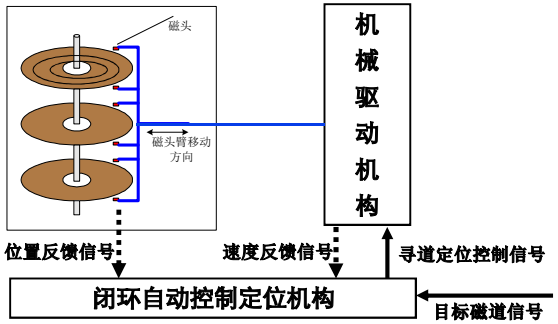
- **硬磁盘存储器的基本结构**
  - 磁盘控制器 ＋ 磁盘驱动器（盘片）
  - 磁盘控制器：将主机发来的命令转化为对驱动器的控制命令，实现主机和驱动器之间的数据格式转化和传送，并控制驱动器读写
  - 磁盘驱动器（盘片）：存储和读写数据



1.2 硬磁盘存储器 —— 基本结构

❖ 磁盘驱动器基本结构

➢存储机构、机械驱动机构、控制机构等



1.2 硬磁盘存储器 —— 基本结构

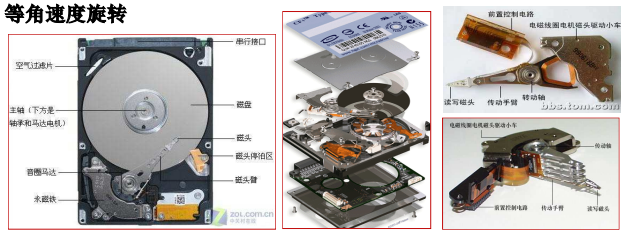
❖ 磁盘驱动器基本结构

➢全密封：浮动磁头组件、磁头驱动机构、盘片和主轴组件和前置控制电路等密封在一起。

➢磁头：非接触式浮动磁头，盘面分启停区和数据区，当磁盘不工作时，磁头停留在启停区表面；磁盘工作时，由于磁盘高速旋转带动气流，使磁头漂浮在磁盘数据区表面上方，头盘间隙仅有0.1微米~0.3微米

➢读写电路：安装在磁头臂接近磁头的地方，以减少干扰

➢旋转速度：3600RPM(转/分)，7200RPM，10KRPM，15KRPM；一般等角速度旋转



1.2 硬磁盘存储器 —— 基本结构

❖ 数据结构与格式

➢数据结构：

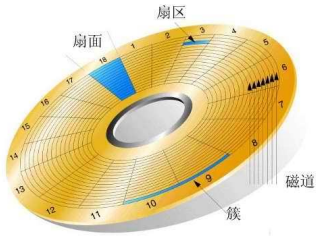
- 盘面（磁头：Head）
- 磁道（柱面：Cylinder）
- 扇区（Sector）

➢每个磁道包含的扇区数相同

➢扇区容量：512 Bytes

➢最小访问单位：扇区

➢扇区的地址表示：



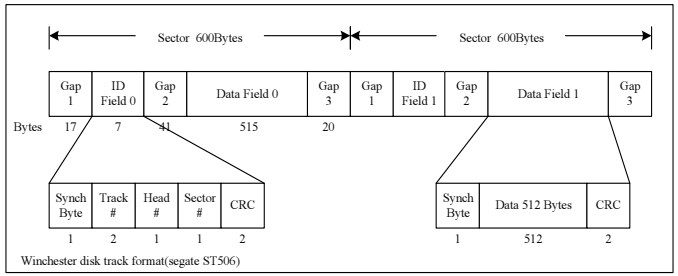
磁盘上的磁道、扇区和簇

扇区地址：

Cylinder#	Head#	Sector#

1.2 硬磁盘存储器 —— 基本结构

❖ 扇区数据格式示例（Segate ST506 磁盘扇区格式）



1.2 硬磁盘存储器 —— 磁盘的性能参数

- 性能指标
  - 记录密度
    - 道密度：磁盘沿半径方向单位长度的磁道数
    - 位密度：单位长度磁道记录二进制的位数
  - 存储容量
    - 盘面数(磁头数) X 每盘面的磁道数 X 每磁道的扇区数 X 扇区容量
  - 访问时间（也称寻址时间）T_A： $T_A=T_S+T_w$
    - 寻道时间 T_S：磁头从当前位置定位到目标磁道所需时间（平均值）
    - 寻区时间 T_w：磁头定位到目标磁道后，等待目标扇区旋转到磁头下所需的时间（平均值）
  - 数据传输率 D_r：单位时间内传输的数据位数（b/s）

1.2 硬磁盘存储器 —— 磁盘的性能参数

- 假设磁盘存储器共有6个盘片，最外两侧盘面不能记录，每面有204条磁道，每条磁道有12个扇段，每个扇区有512byte，磁盘以7200rpm速度旋转，平均寻道时间为8ms

  计算该磁盘存储器的：

  （1）容量

  （2）平均寻址时间

  （3）数据传输率

- 解答：

（1）容量： （6x2-2）X 204 X 12 X 512 = 12533760 bytes

（2）平均寻址时间： 8 + （60*1000 / 7200）/ 2 = 12.165 ms

（3）数据传输率：（7200 / 60）X 12 X（512x8） = 5898240 b/s

第八讲：虚拟存储系统

1.3 磁盘的类型

- 磁盘的类型
  - Floppy Disk
  - Hard Disk
    - IDE（Integrated Drive Electronics）/ EIDE
    - SCSI（Small Computer System Interface）
    - SATA（Serial-ATA）
    - SAS（Serial Attached SCSI）

1.3 磁盘的类型 —— 软盘

❖ 软盘（Floppy Disk）

➢尺寸：5.25 inch, 3.5 inch

➢容量：360KB, 1.2MB, 720KB, 1.44MB

1.3 磁盘的类型 —— 硬盘

❖ IDE（Integrated Drive Electronics）磁盘

- 80年代出现，最初为AT结构的PC设计，又称ATA(Advanced Technology Attachment)接口，属低价位磁盘，由系统BIOS（Basic Input Output System)处理磁盘的读写等操作
- 早期可能是BIOS程序员的失误，IDE磁盘的地址被定义为：Head #(4位，从0开始)，Cylinder#(10位，从0开始)，Sector#(6位，从1开始)，所以磁盘最大容量限制：16*63*1024*512Bytes（528MB）
- 后出现了EIDE（Extendex IDE）Hard Disk，支持LBA (Large Block Address)地址模式，扇区地址可从0到2^{24}-1
- IDE接口用一根40芯或80芯的扁平电缆连接硬盘与主板，每条线最多连接2/4个IDE设备。Ultra DMA/133其数据传输率为133 MB/s，对CPU的占用率从92%降至52%

1.3 磁盘的类型 —— 硬盘

❖ SCSI（Small Computer System Interface）磁盘：数据结构和磁盘结构与IDE类似。但有不同的接口和更高的数据传输率

- 50 wires cable (8-bit)：GND(25 wires), Data(8 wires), Parity (1 wire),Control(9 wires), Power (Others)
- SCSI总线是共享的，所有SCSI设备（不一定是磁盘）可以同时操作，这是与IDE/EIDE最大的不同之处
- SCSI接口速度可达320MB/s，硬盘容量可达300GB

Name	Data bits	Bus Mhz	MB/Sec
SCSI-1	8	5	5
SCSI-2	8	5	5
Fast SCSI-2	8	10	10
Fast & Wide SCSI-2	16	10	20
Ultra SCSI	16(32)	20	40
Ultra2 Wide SCSI	16		80
Ultra-160m/Ultra-320m	16		160/320

1.3 磁盘的类型 —— 硬盘

❖ SATA（Serial-ATA）磁盘：IDE属Parallel-ATA，SATA具有更快的外部接口传输速度，数据校验措施更为完善

- 改用线路相互间干扰较小的串行线路进行信号传输，相比原来的并行总线，SATA的工作频率得以大大提升
- SATA具有更简洁方便的布局连线方式，在有限机箱内，更有利于散热，使内部电磁干扰降低很多
- SATA 1.0速率可达150MB/s，硬盘容量可达400GB。SATA 2.0/3.0可提升到300至600MB/s
- SATA是点对点的，但SCSI总线是共享的

1.3 磁盘的类型 —— 硬盘

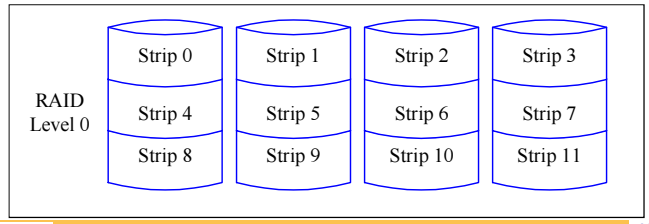
- SAS（Serial Attached SCSI）即串行连接SCSI，是新一代SCSI技术，SAS改善了存储系统的效能、可用性和扩充性
  - 与SATA硬盘相同，都采用串行技术以获得更高的传输速度，并通过减少连结线改善内部空间等
  - 由于采用串行线缆，可实现更长的连接距离，还能提高抗干扰能力，且可显著改善机箱内部的散热情况
  - SAS起步速度可达300MB/s，600MB/s甚至更多
  - SAS是点对点的，SAS的接口技术可向下兼容SATA
  - 支持厂商（希捷、迈拓、富士通）较少，SAS相关的硬盘、控制芯片种类少

1.3 磁盘的类型 —— RAID

- RAID
  - Reduntant Array of Inexpensive Disks（廉价磁盘冗余阵列）
    - RAID由多个物理磁盘构成，但被操作系统当成一个逻辑磁盘，数据分布在不同的物理磁盘上，冗余磁盘用于保存数据校验信息，校验信息保证在出现磁盘损坏时能够有效地恢复数据
  - RAID特点
    - 通过把多个磁盘组织在一起作为一个逻辑卷提供磁盘跨越功能
    - 通过把数据分成多个数据块（Block）并行写入/读出多个磁盘以提高访问磁盘的速度
    - 通过镜像或校验操作提供容错能力
  - 不同模式的RAID
    - RAID 0，RAID1, RAID2，RAID3，RAID4，RAID 5

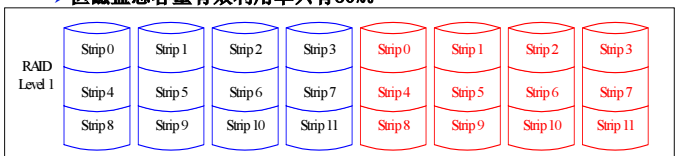
1.3 磁盘的类型 —— RAID

- RAID 0：无差错控制的带区组
  - 实际上不应属于RAID家族成员，完全没有冗余；
  - 数据条带化（Strip）地分布在不同的物理磁盘上。Strip可以是物理磁盘上的一块存储区（扇区或其他单位）。
  - 磁盘组中每一个磁盘同一位置的磁盘区构成一个逻辑上的条带，所以一个条带分布在多个磁盘上。
  - 单个I/O 操作访问的数据分布在一个条带上是，可实现I/O操作的并行处理，改善数据传输性能。



1.3 磁盘的类型 —— RAID

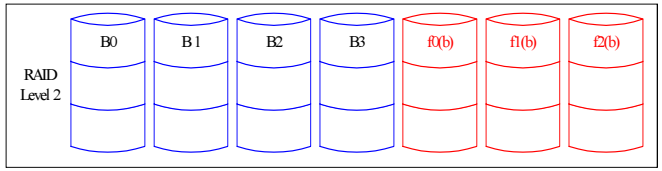
- RAID 1：镜像结构
  - 简单镜像磁盘冗余方案，成本太高；
  - 与RAID 0类似，用户数据和系统数据条带化（Strip）地分布在不同的物理磁盘上（包括镜像磁盘）。
  - 读操作同时在两组磁盘中进行，数据从访问时间小的磁盘组中获得，所以，读操作性能得到改善。
  - 写操作同时在两组磁盘中进行，写操作的访问时间以速度慢的为准，所以，写操作性能指标不高。
  - 出现磁盘损坏时，数据恢复简单。
  - 但磁盘总容量有效利用率只有50%。



1.3 磁盘的类型 —— RAID

❖ RAID 2：带海明校验

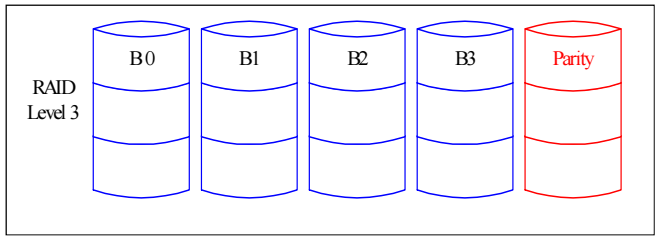
- 采用完整的并行访问技术，所有磁盘在任何时刻都并行地响应I/O 请求；磁盘组中物理磁盘处于完全同步状态，以保证任何时刻，所有磁盘的磁头都处于相同位置。
- 数据按较小的条带（一个字或一个字节）分布在不同的磁盘上。
- 根据磁盘数据计算错误校验码（比如海明码），校验码按位分布在冗余磁盘对应位置上。
- 数据传输率高；访问效率高；
- 成本比较高（比RAID1稍低）



1.3 磁盘的类型 —— RAID

❖ RAID 3：带奇偶校验码的并行传送

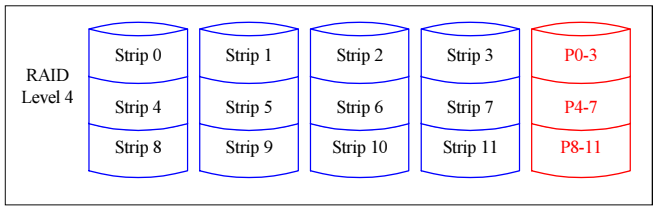
- 与RAID2一样，采用并行访问技术；
- 数据按较小的条带（一个字或一个字节）分布在不同的磁盘上。
- 校验码是简单的奇偶校验码（1位），保存在独立的冗余磁盘对应位置上。
- 一个磁盘损坏，可以方便地实现数据恢复；
- 数据传输率高；访问效率高；



1.3 磁盘的类型 —— RAID

❖ RAID 4：带奇偶校验码的独立磁盘结构

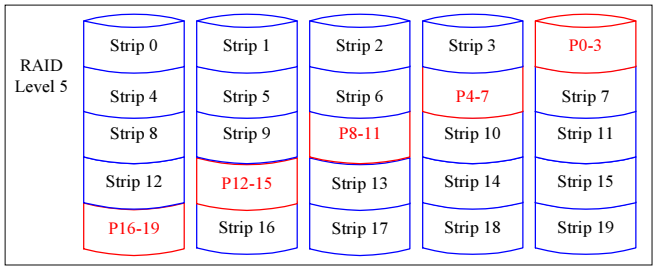
- 采用独立访问技术，每个磁盘独立工作，分散的I/O请求将得到很好的并行处理
- 数据按较大的条带分布在不同的磁盘上。
- 校验码是奇偶校验码，保存在独立的冗余磁盘对应位置上。
- 一个磁盘损坏，可以方便地实现数据恢复；
- 写操作效率较低，需计算奇偶校验位，磁盘组中一个磁盘写操作，均需要读取原检验信息，重新计算校验信息，再写校验信息。



1.3 磁盘的类型 —— RAID

❖ RAID 5：分布式奇偶校验的独立磁盘结构

- 与RAID 4的差别仅在于校验信息的保存位置；数据校验码作为条带的一部分保存在磁盘组不同的磁盘中

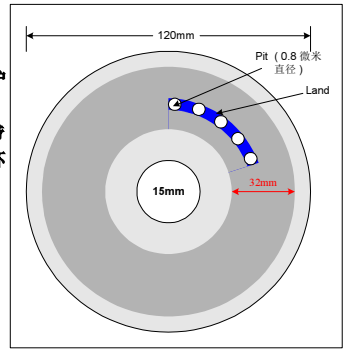


第八讲：虚拟存储系统

一. 辅助存储器
1. 磁记录方式
2. 硬磁盘存储器
3. 磁盘的类型
4. 光盘存储器
二. 虚拟存储器

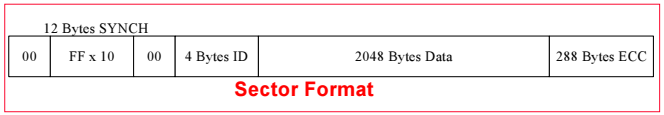
1.4 光盘存储器 —— CD-ROM

- CD-ROM
  - 规格：直径120mm，厚1.2mm，中心孔径 15mm
  - 结构：树脂片基，铝反射层，保护膜，印刷层
  - 数据记录区：32mm宽环形记录带
    - 等线速度方式：一个螺旋环环绕22188次（600环/mm，总长度约 5.6km长）
    - 等角速度方式
- 数据记录
  - 凹点（Pit）表示 0，Land 表示 1
  - 制作过程：母板压模
- 读机制：0.78微米波长红外激光，根据反射光的强度判断是0还是1



1.4 光盘存储器 —— CD-ROM

❖ CD-ROM的数据格式

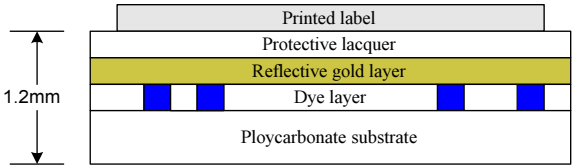


- Symbol：14位，8位数据，6位海明校验位(看成一个Bytes)；
- Frame：42个连续Symbol（588bits），其中192位（24字节）存储数据，其余396位用于错误纠正与控制
- Sector：98个frame构成一个Sector（总计2352Bytes）
- 总容量：650MB
- 等线速度旋转时：单速：120cm/s （最内圈530RPM，最外圈200RPM），75 Sectors/Sec（150KB/S）

1.4 光盘存储器 —— CD-R

❖ CD-R（Recordables）

- 在片基（树脂）与反射层（金）中增加了一层染料层作为数据记录层，初始状态下，染料层透明，在写入状态时，高能量（8-16mw）使照射处的染料变色，变成不透明点，不可再恢复成透明状态。读出状态下(0.5mw)，根据透明不透明判断是0还是1



1.4 光盘存储器 —— CD-RW

❖ CD-RW（Rewritables）

- 与CD-R的差别是采用合金层代替染料层。一般采用银、铟、锑、碲合金。该合金具有两种稳定状态：透明状态（晶体结构）和不透明状态（无序结构），初始时为晶体结构。
- CD-RW工作时采用三种不同功率的激光：
  - 大功率（写）：合金熔化，由晶体结构变为无序结构
  - 中等功率（擦除）：合金熔化，由无序结构变为晶体结构
  - 小功率（读）

1.4 光盘存储器 —— DVD

❖ DVD（Digital Video Disk）

与CD-ROM的差别：

- Pit直径更小（0.4微米）
- 环绕密度更高（0.74微米，CDROM是1.6微米）
- 0.65微米波长红色激光（CDROM是0.78微米的红外激光）
- 容量：单面单层4.7GB，单面双层8.5GB，双面单层9.4GB，双面双层17GB
- 数据传输率：单速DVD 1.4M Bytes/Sec

第八讲：虚拟存储系统

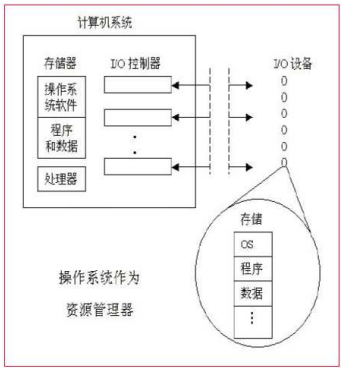
一. 辅助存储器

二. 虚拟存储器

1. 虚拟存储器概述
2. 页式虚拟存储器

2.1 虚拟存储器概述

- 存储器管理：操作系统的功能
- 操作系统：合理地管理、调度计算机的硬件资源。存储器作为一种空间资源也由OS来管理；
- CPU执行的程序：总是在操作系统和用户程序之间切换。主存中同时要存储OS和用户程序。磁盘中也要存储OS和用户程序；
- CPU中的存储器管理部件MMU协助OS完成存储器访问。



2.1 虚拟存储器概述

■ 一个典型程序的转换处理过程

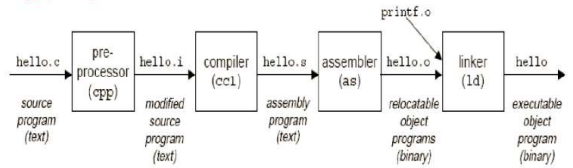
经典的" hello.c "C-源程序

```
1 #include <stdio.h>
2
3 int main()
4 {
5 printf("hello, world\n");
6 }
```

程序的功能是：
输出"hello,world"

hello.c的ASCII文本表示

```
# i n c l u d e <sp> < s t d i o .
35 105 110 99 108 117 100 101 32 60 115 116 100 105 111 46
h > \n \n i n t <sp> m a i n ( ) \n {
104 62 10 10 105 110 116 32 109 97 105 110 40 41 10 123
\n <sp> <sp> <sp> <sp> p r i n t f ( " h e l
10 32 32 32 32 112 114 105 110 116 102 40 34 104 101 108
l o , <sp> w o r l d \ n " ) ; \n }
108 111 44 32 119 111 114 108 100 92 110 34 41 59 10 125
```



2.1 虚拟存储器概述

■ 操作系统在程序执行过程中的作用

- Hello执行过程中，其本身没有直接访问键盘、显示器、磁盘和主存储器这些硬件资源，而是依靠操作系统提供的服务来间接访问。
- 操作系统的两个主要的作用：
  - 硬件资源管理
  - 为用户使用系统提供一个操作接口
- 操作系统通过三个基本的抽象概念（进程、虚拟存储器、文件）实现硬件资源管理
  - 文件（files）是对I/O设备的抽象表示
  - 虚拟存储器（Virtual Memory）是对主存和磁盘I/O的抽象表示
  - 进程（processes）是处理器、主存和I/O设备的一种抽象表示，实际上是对运行程序的抽象表示

2.1 虚拟存储器概述

■ 虚拟存储器的动机

- 多道程序（进程）同时运行时如何共享存储器？
  - 同时运行的程序对内存的需求之和可能超过计算机实际内存容量
  - 单个程序对内存的需求也有可能超过机器实际内容容量
- 如何消除小的主存容量对编程的限制？
  - 编写编译程序时，不知道程序运行时将和哪些其他程序共享内存
  - 我们希望把每个程序编译在它自己的地址空间中

■ 解决之道

- 程序运行时，内存管理采用交换机制（硬件和操作系统实现），进程保存在辅存中，进程执行时，只将其活跃部分调入内存（局部性原理）。此时主存可以视为辅存的“高速缓存”
- 这样一种把主存当做辅助存储器的高速缓存的技术，称为虚拟存储器（virtual memory）技术

2.1 虚拟存储器概述

■ 虚存空间与物理空间

- 用户编程空间：用户编制程序时使用的地址称为**虚地址或逻辑地址**，其对应的存储空间称为**虚存空间**或逻辑地址空间。虚存空间的用户程序按照虚地址编程并存放在辅存中。
- 物理内存空间：计算机物理内存的访问地址称为**实地址**或**物理地址**，其对应的存储空间称为**物理空间**或主存空间。

■ 虚拟存储器要解决的问题

- 虚存空间与物理空间之间的数据交换：交换哪些数据？每次交换多少？
- 虚地址与实地址的转换问题：虚地址格式、实地址格式、怎么判断当前访问的虚地址对应的数据是不是在物理空间中，如何把虚地址转换为实地址？如何加速这种判断和转换？
- 缺失处理和替换策略：访问的内容不在物理空间中怎么处理？

2.1 虚拟存储器概述

■ 虚拟存储系统的特征

- 程序员在比实际主存空间大得多的逻辑地址空间中编写程序
- 程序执行时，把当前需要的程序段和相应的数据块调入主存，其他暂不用的部分存放在磁盘上
- 指令执行时，通过硬件（MMU）将逻辑地址（也称虚拟地址或虚地址）转化为物理地址（也称主存地址或实地址）
- 在发生程序或数据访问失效时，由操作系统进行主存和磁盘之间的信息交换。

虚拟存储器机制由硬件与操作系统共同协作实现，涉及到操作系统中的许多概念，如进程、进程的上下文切换、存储器分配、虚拟地址空间、缺页处理等。

2.1 虚拟存储器概述

■ 虚拟存储器的调度方式

- **页式调度**：将虚存空间和物理地址空间都分成固定大小的页。主存按页顺序编号；每个独立编址的程序空间也按自己的页顺序编号。虚存空间和物理空间按页进行交换。
- **段式调度**：按程序的逻辑结构将程序空间划分为若干段，段的长度是随意的，虚存空间和物理空间按段进行交换。
- **段页式调度**：两种方法的结合。在段页式调度中把物理空间分成页，程序按模块分段，每个段再分成与物理空间页同样大小的页面。虚存空间和物理空间按页进行交换。

2.1 虚拟存储器概述

■ 虚拟存储器小知识

1. 虚拟存储器源自于英国ATLAS计算机（1962年，英国曼彻斯特大学）的一级存储器概念。这种系统的主存为16千字的磁芯存储器，但中央处理器可用20位逻辑地址对主存寻址。成为现在广为采用的虚拟存储器的雏形。
2. 1970年，美国RCA公司研究成功虚拟存储器系统
3. 1972年，IBM公司于在IBM370系统上全面采用了虚拟存储技术
4. 当前，虚拟存储器已成为计算机系统中非常重要的部分

第八讲：虚拟存储系统

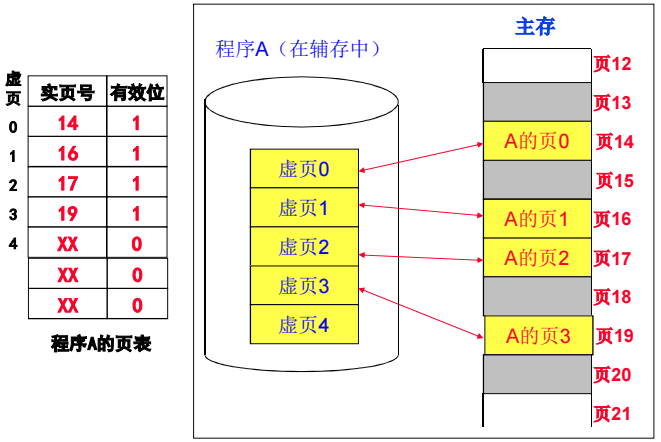
2.2 页式虚拟存储器

- **页式虚拟存储器**
  - 主存分成固定长度且比较小的存储块，称为实页（物理的页）；
  - 进程也划分成相同长度的程序块，称为虚页（虚拟的页）；
  - 主存按页顺序编号，每个独立编址的程序（进程）空间有自己的页号顺序，通过调度辅存中程序各页可离散装入主存不同实页位置。
  - CPU执行指令时，首先需将逻辑地址转换为主存的物理地址，地址转换由CPU中的MMU实现。
  - 页式调度：按页交换
    - 优点：页内零头小，页表对程序员来说是透明的，地址变换快，调入操作简单；
    - 缺点：各页不是程序的独立模块，不便于实现程序和数据保护。

2.2 页式虚拟存储器

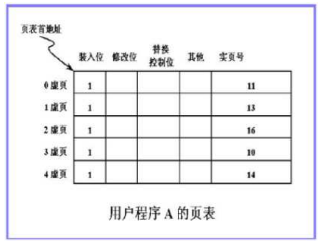
- **页式虚拟存储器**
  - 虚存空间和主存空间按固定大小分成若干页，虚存页称为虚页，主存页称为实页。辅存中的程序按页调入内存
  - 虚地址格式（逻辑地址格式）：虚页号 ＋ 页内地址
  - 实地址格式（物理地址格式）：实页号 ＋ 页内地址
  - 页表：记录虚页与实页的映射关系，实现虚实地址的转换，页表建立在内存中，操作系统为每道程序建立一个页表。页表用虚页号作为索引，页表项包括虚页对应的实页号和有效位
  - 页表寄存器：保存页表在内存中的首地址。

2.2 页式虚拟存储器



2.2 页式虚拟存储器

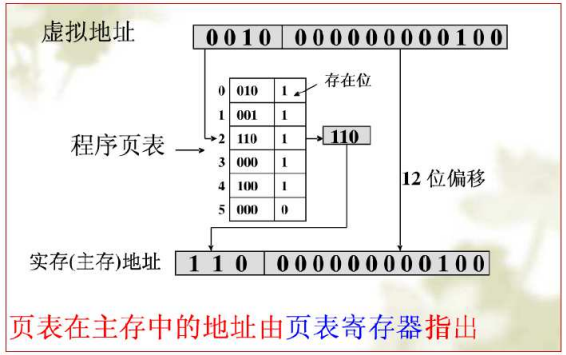
- **页表**
  - 每个进程有一个页表，页表项数取决于虚拟地址的结构。
  - 页表项包括：实页号、装入位、修改位等
  - 页表在内存中的首地址记录在页表基址寄存器中。



- **页表空间问题**
  - 页表可能很大。如VAX系统中，用户程序（进程）虚拟存储空间最大可达2GB，按512B分页，有2^{22}页，页表可以包括2^{22}个页表项。显然，这么大的页表需要占用很大的内存空间。
  - 多进程运行，对个页表同时都在内存。多个同时运行的进程的页表空间超过内存可分配空间的可能性是存在的。
  - 采用多级页表机制：将页表分页，当前使用的页的页表项所在页在内存，其余在外存，页表也采用按页交换机制。

2.2 页式虚拟存储器

■ 虚拟地址到物理地址的转换



2.2 页式虚拟存储器

❖ 举例

某计算机虚拟地址32位，物理内存128MB，页大小4KB。

（1）程序虚拟空间最多可有多少页？

（2）页表项共有多少位？

（3）每个页表占多少内存空间？

❖ 解答

虚地址32位：虚页号（20位）+ 页内偏移（12位）

实地址27为：实页号（15位）+ 页内偏移（12位）

每个程序虚拟空间最多可有：2^{20}个虚页

每个页表项：1位（有效位）+ 15位（实页号）= 16位

每个页表所占空间：$2^{20} \times 16$ = 16Mb = 2MB

多道程序运行时，所有程序的页表都在内存中，页表占用内存空间不可忽视，极端情况下，页表有可能消耗所有内存空间。（采用多级页表解决）

2.2 页式虚拟存储器

■ 快表TLB（Translation Lookaside Buffer，转换后备缓冲器）

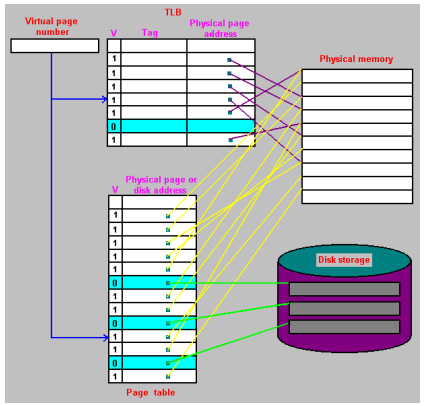
- 问题：每次虚拟存储器的访问带来两次存储器访问，一次访问页表，一次访问所需的数据（或指令），简单的虚拟存储器速度太慢
- 解决办法： 使用Cache存储部分活跃的页表项，称为TLB（快表），它包含了最近使用的那些页表项
- TLB内容：虚页号（标记）、对应实页号（数据）、有效位、修改位
- TLB一般采用全相联模式

有效位	修改位	标记（tag）	数据
		虚页号	实页号
		虚页号	实页号
		虚页号	实页号
		虚页号	实页号

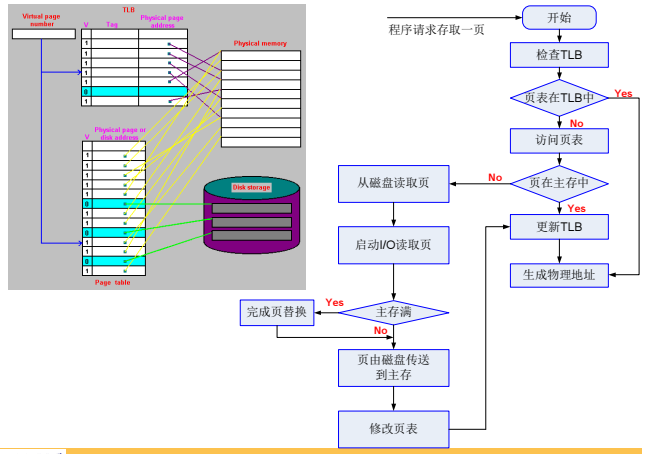
快表（TLB）

2.2 页式虚拟存储器

■ 页式虚拟存储器工作原理



2.2 页式虚拟存储器



2.2 页式虚拟存储器

■ TLB，页表，Cache三种缺失的可能性

TLB	Page table	Cache	Possible? If so, under what circumstance?
hit	hit	miss	可能，TLB命中则页表一定命中，但实际上不会查页表
miss	hit	hit	可能，TLB缺失但页表可能命中，信息在主存，就可能在Cache
miss	hit	miss	可能，TLB缺失但页表可能命中，信息在主存，但可能不在Cache
miss	miss	miss	可能，TLB缺失页表可能缺失，信息不在主存，一定也不在Cache
hit	miss	miss	不可能，页表缺失，说明信息不在主存，TLB中一定没有该页表项
hit	miss	hit	同上
miss	miss	hit	不可能，页表缺失，说明信息不在主存，Cache中一定也没有该信息

最好的情况应该是hit、hit、hit，此时，访问主存几次？ 不需要访问主存！

以上组合中，最好的情况是什么？hit、hit、miss和miss、hit、hit 只需访问主存1次

以上组合中，最坏的情况是什么？ miss、miss、miss 需访问磁盘、并访存至少2次

介于最坏和最好之间的是什么？ miss、hit、miss 不需访问磁盘、但访存至少2次

举例

■ 假定页式虚拟存储系统按字节编址，逻辑地址36位，页大小16KB，物理地址32位，页表中包括有效位和修改位各1位、使用位和存期方式位各2位，且所有虚拟页都在使用中。请问：

- （1）每个进程的页表大小为多少？
- （2）如果所使用的快表（TLB）总表项数为256项，且采用2路组相联Cache实现，则快表大小至少为多少？

解答（1）

页面大小：16KB=214，页内偏移14位
虚地址36位：虚页号=36-14=22位
实地址32为：实页号=32-14=18位
每个进程最多可有：222个虚页
每个页表项：1+1+2 +2+ 18=24位
每个页表所占空间：222×24=12MB

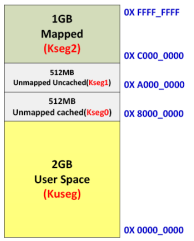
（2）

TLB：256个表项，2路组相联，所以共有128组
22位虚页号：7位组地址，15位Tag
TLB每个表项：15+24=39位
TLB容量：39×256=9984位=1248字节

MIPS的存储空间管理

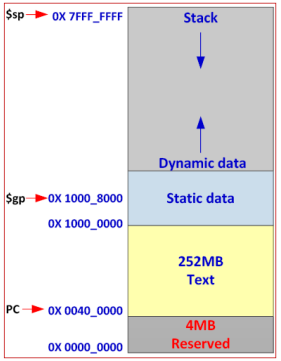
■ MIPS CPU运行模式：用户态和核心态

- Kuseg：0000000 -7FFF FFFF (2G)，用户模式下可用，即MIPS规范约定用户空间为2G在带有MMU的机器里，这些地址都将由MMU转换。
- KSeg0：80000000 -9FFFFFFF(512M)，通过将最高位清零映射到物理地址低端512M(00000000 -1FFF FFFF)空间。这种映射简单，无需MMU转换（Unmapped）。这段地址的存取都会通过高速缓存(cached)，对于有MMU 的系统，操作系统内核会存放在该区域。
- KSeg1：A0000000 -BFFFFFFF(512M)，将最高3位清零映射到物理地址低端512M(00000000 -1FFFFFFF)空间，无需MMU转换（Unmapped），kseg1不使用缓存（Uncached）。kseg1是系统重启时能正常工作的内存映射地址空间，重新启动时的入口向量是BFC00000，这个向量对应的物理地址是1FC00000。因此你可以使用这段地址空间来访问你的初始化程序的ROM。
- KSeg2：C0000000 -FFFFFFFF (1G)，只能在核心态下使用，并且要经过MMU转换，一般情况下你不需要使用这段地址空间。



MIPS的存储空间管理

- MIPS按如下约定为程序分配空间
  - 堆栈在高地址区，从高到低增长。
  - 过程调用时，生成当前“栈帧”，返回后退回当前栈帧
  - 程序的动态数据（如：C中的malloc申请区域、链表）在堆(heap)中从低向高进行存放和释放（free时）栈区位于堆栈高端，堆区位于堆栈低端，静态数据区上方。
  - 全局指针$gp固定设为0x10008000，其16位偏移量的访问范围为0x10000000 到 0x1000FFFF
  - 静态数据区从固定的0x10000000处开始存放
  - 程序代码从固定的0x00400000处开始存放，故PC的初始值为0x00400000。



MIPS每个进程的虚拟（逻辑）地址空间